Xerox Palo Alto Research Center Technical Report P2001-10113, May 2001. Revised July 25, 2001.

Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks

Maurice Chu, Horst Haussecker, and **Feng Zhao***
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304, U.S.A.

Abstract

This paper describes two novel techniques, information-driven sensor querying (IDSQ) and constrained anisotropic diffusion routing (CADR), for energy-efficient data querying and routing in ad hoc sensor networks for a range of collaborative signal processing tasks. The key idea is to introduce an information utility measure to select which sensors to query and to dynamically guide data routing. This allows us to maximize information gain while minimizing detection latency and bandwidth consumption for tasks such as localization and tracking. Our simulation results have demonstrated that the information-driven querying and routing techniques are more energy efficient, have lower detection latency, and provide anytime algorithms to mitigate risks of link/node failures.

1 Introduction

Advances in wireless networking, microfabrication (e.g. MEMS), and embedded microprocessors have enabled a new generation of large sensor networks potentially applicable to a range of tracking and identification problems in both civilian and military applications. Examples include human-aware environments, intelligent transportation grids, factory condition-based monitoring and maintenance, and battlefield situational awareness. With such advances come new challenges for information processing which necessitate the development of novel routing and collaborative signal processing (CSP) algorithms to handle the distributed data gathered by the network and process the information for the given task.

Distributed sensor networks are characterized by limited battery power, frequent node attrition, and variable data and communication quality. To scale up to realistic tracking and classification applications involving tens of thousands of sensors, heterogeneous sensing modalities, multiple targets, and non-uniform spatio-temporal scales, these systems have to rely primarily on intelligent collaboration among distributed sensors to significantly improve tracking accuracy and reduce detection latency.

Since sensors often must collaborate in order to perform a given sensing task about the environment, determining the utility of a remote sensor's data is important in conserving power due to the necessity of communication. Thus, the problem we are addressing in this paper is *how to dynamically query sensors and route data in a network so that information gain is maximized while latency and bandwidth consumption is minimized*. Our approach relies on two *key ideas*: information driven sensor querying (IDSQ) to optimize sensor selection and constrained anisotropic diffusion routing (CADR) to direct data routing and incrementally combine sensor measurements so as to minimize an overall cost function. Dynamic sensor tasking and collaboration has the following advantages. It is key to the scalability of large-scale sensor networks through selective sensor tasking. It can drastically reduce latency in detection and tracking by application-aware optimal routing. It enables progressive accuracy by incremental belief update. It ensures graceful degradation in performance due to link failure or sensor attrition by providing anytime algorithms.

Using information utility measures to optimize sensor selection and incremental belief update to combine information in a centralized system is not new (e.g., active testing [Geman&Jedynak 1996] and Kalman filter), nor is cost-based data routing in ad hoc networks (e.g., directed diffusion routing [Intanagonwiwat et al. 2000]). What is new about IDSQ/CADR is the use of a general form of information utility that models the information content as well as the spatial configuration of a network in a distributed way; with this formulation, each node can evaluate an information/cost objective, make a decision, update its belief state, and route data based on the local information/cost gradient and end-user requirement. Existing approaches to sensor querying and data routing for ad hoc heterogeneous sensor networks typically employ a publish-subscribe scheme [Huang&Garcia-Molina 2001], where publishers advertise their data "attributes", subscribers submit "interests", and an event brokering system matches "interests" with "attributes". The directed diffusion [Intanagonwiwat et al. 2000] is one such example, where the routing paths are established using the distance information between neighboring nodes to minimize the number of hops in RF communication. IDSQ/CADR can be considered

*Corresponding author. Email: zhao@parc.xerox.com, Tel: 650-812-5078.

as a generalization of directed diffusion routing. We use both information gain and communication cost to direct the data diffusion, and hence can be more energy aware and efficient than directed diffusion routing where typically only communication cost is of concern.

Our approach allows sensors to become activated when there are "interesting" events to report, and only those parts of the network with the most useful information balanced by the communication cost need to be active. The type of networks can be stealthy and is advantageous for security reasons. The networks could also actively seek out information, based on predictions of when and where "interesting" events will be. An entire sensor network, with its in-network processing power for processing, routing, and combining distributed sensor data, is an extremely powerful distributed computing system. One notable feature of this distributed "supercomputer" is that its I/O is also distributed and can match well with the impedance of the sensing application domain by design.

The rest of the paper is structured as follows. Section 2 introduces a general formulation of the problem in terms of an information utility measure and mentions issues of representing and incrementally updating the belief state. Section 3 presents the algorithms for information driven sensor querying and constrained anisotropic diffusion routing. Section 4 describes experimental evidence and compares our approach with existing work in uncertainty management and data diffusion routing. Section 5 elaborates on other routing protocols and issues of belief representation, and Section 6 discusses related work.

2 Problem Formulation

2.1 Sensing Model and Measure of Uncertainty

We will model our estimation problem using standard estimation theory. The time-dependent measurement, $\mathbf{z}_i(t)$, of sensor i with characteristics, $\lambda_i(t)$, is related to the parameters, $\mathbf{x}(t)$, that we wish to estimate through the following observation (or measurement) model

$$\mathbf{z}_i(t) = \mathbf{h}(\mathbf{x}(t), \lambda_i(t)), \tag{1}$$

where $\mathbf{h}$ is a (possibly non-linear) function depending on $\mathbf{x}(t)$ and parameterized by $\lambda_i(t)$, which represents the (possibly time dependent) knowledge about sensor i. Typical characteristics, $\lambda_i(t)$, about sensor i include sensing modality, which refers to what kind of sensor i is, sensor position $\mathbf{x}_i$, and other parameters, such as the noise model of sensor i and node power reserve. The reason for explicitly representing these sensor characteristics is due to the distributed and heterogeneous nature of the sensor processing tasks.

In (1), we consider a general form of the observation model that accounts for possibly non-linear relations between the sensor type, sensor position, noise model, and the parameters we wish to estimate. A special case of (1) would be

$$\mathbf{h}(\mathbf{x}(t), \lambda_i(t)) = \mathbf{f}_i(\mathbf{x}(t)) + \mathbf{w}_i(t),$$

where $\mathbf{f}_i$ is a (possibly non-linear) observation function, and $\mathbf{w}_i$ is additive, zero mean noise with known covariance.

In case $\mathbf{f}_i$ is a linear function on the parameters, (1) reduces to the linear equation

$$\mathbf{h}(\mathbf{x}(t), \lambda_i(t)) = \mathbf{H}_i(t)\,\mathbf{x}(t) + \mathbf{w}_i(t), \tag{2}$$

In order to illustrate our technique, we will later consider the problem of stationary target localization with stationary sensor characteristics. Here, we assume that all sensors are acoustic sensors measuring only the amplitude of the sound signal so that the parameter vector $\mathbf{x} = [x, y]^T$ is the unknown target position, and

$$\lambda_i = \left[\mathbf{x}_i, \sigma_i^2\right]^T, \tag{3}$$

where $\mathbf{x}_i$ is the known sensor position, and σ_i^2 is the known additive noise variance. Note there is no longer a time dependence for $\mathbf{x}$ and λ_i. Assuming that acoustic signals propagate isotropically, the parameters are related to the measurements by

$$z_i = \frac{a}{\|\mathbf{x}_i - \mathbf{x}\|^{\frac{\alpha}{2}}} + w_i \tag{4}$$

where a is a given random variable representing the amplitude of the target, α is a known attenuation coefficient, and $\|\cdot\|$ is the Euclidean norm. w_i is a zero mean Gaussian random variable with variance σ_i^2.

In the remainder of this paper, we define the *belief* as a representation of the current *a posteriori* distribution of $\mathbf{x}$ given measurements $\mathbf{z}_1, \ldots, \mathbf{z}_N$:

$$p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)\ .$$

Typically, the expectation value of this distribution

$$\bar{\mathbf{x}} = \int \mathbf{x}\, p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)\, d\mathbf{x}$$

is considered the estimate (i.e., the minimum mean square estimate), and we approximate the residual uncertainty by the covariance:

$$\Sigma = \int (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)\, d\mathbf{x}.$$

What standard estimation theory does not consider and is of great importance to distributed sensor networks, however, is that knowledge of the measurement value $\mathbf{z}_i$ and sensor characteristics λ_i normally resides only in sensor i. In order to compute the belief based on measurements from several sensors, we must pay a cost for communicating that information. Thus, maintaining what information each sensor node has about other sensor nodes is an important decision. This is why the sensor characteristics $\lambda_i(t)$ are explicitly represented because it is important to know what information is available for various information processing tasks.

Since incorporating measurements into the belief are now assigned costs, the problem is to intelligently choose a subset of sensor measurements which provide "good" information for constructing a belief state as well as minimizing the cost of having to communicate sensor measurements to a single node. In order to choose sensors to provide "good" updates to the belief state, it is necessary to introduce a measure of the information a sensor measurement can provide to a belief state. We will refer to this as the *information content* of a

particular sensor. The formalization of the criterion for selecting the next best sensor is the main contribution of this paper. Section 2.2 will make this intuitive notion of information content mathematically rigorous, Section 2.3 will propose several instances of information content measures that can be practically implemented, and Section 2.4 will combine the measures with communication costs to provide a composite utility function.

2.2 Sensor Selection

In Section 2.1 we formulated the distributed sensing model. Given the current belief state, we wish to incrementally update the belief by incorporating measurements of previously not considered sensors. However, among all available sensors in the network, not all provide useful information that improves the estimate. Furthermore, some information might be useful, but redundant. The task is to select an optimal subset and to decide on an optimal order of how to incorporate these measurements into our belief update.

Figure 1 illustrates the basic idea of optimal sensor selection. The illustration is based upon the assumption that estimation uncertainty can be approximated by a Gaussian distribution, illustrated by uncertainty ellipsoids in the state space. In both figures, the solid ellipsoid indicates the belief state at time t, and the dashed ellipsoid is the incrementally updated belief after incorporating an additional sensor at the next time step. Although in both cases, *a* and *b*, the area of high uncertainty is reduced by the same amount, the residual uncertainty in case *a* maintains the largest principal axis of the distribution. If we were to decide between the two sensors, we might favor case *b* over case *a*, based upon the underlying measurement task.

It has to be emphasized that, due to the distributed nature of the sensor network, this selection has to be done without explicit knowledge of the measurement residing at each individual sensor, in order to avoid communicating useless information. Hence, the decision has to be made solely based upon the sensor characteristics λ_i, and the predicted contribution of these sensors.

Although details of the implementation depend on the network architecture, the fundamental principles derived in this section hold for both, the selection of a remote sensor by a cluster head, as well as the decision of an individual sensor to contribute its data and to respond to a query traveling through the network. The task is to select the sensor that provides the best information among all available sensors that have not been incorporated. As will be shown in the experimental results, this provides a faster reduction in estimation uncertainty, and usually incurs lower communication overhead for meeting a given estimation error requirement, compared to blind or nearest-neighbor sensor selection schemes.

To derive a mathematical formulation of the sensor selection process we assume there are N sensors labeled from 1 to N and the corresponding measured values of the sensors are $\{\mathbf{z}_i\}_{i=1}^{N}$. Let $U \subset \{1, \ldots, N\}$ be the set of sensors whose measurements have been incorporated into the belief. That is, the current belief is

$$p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U}) \ .$$

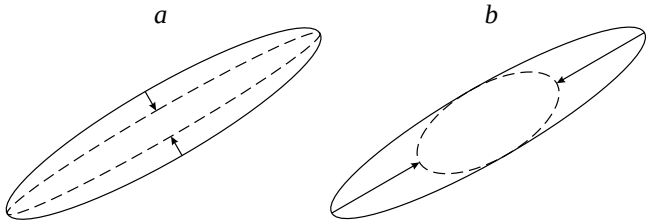

Figure 1: Illustration of sensor selection criteria based upon the information gain of individual sensor contributions. Here, the information gain is measured by the reduction in the error ellipsoid.

The sensor selection task is to choose a sensor which has not been incorporated into the belief yet which provides the most information. To be precise, let us define an information utility function

$$\psi : \mathcal{P}(\mathcal{R}^d) \rightarrow \mathcal{R}$$

which acts on the class $\mathcal{P}(\mathcal{R}^d)$ of all probability distributions on R^d and returns a real number with d being the dimension of $\mathbf{x}$. The role of ψ is to assign a value to each element $p \in \mathcal{P}(\mathcal{R}^d)$ which indicates how spread out or uncertain the distribution p is. Smaller values represent a more spread out distribution while larger values represent a tighter distribution. We will defer discussion of different choices of ψ to Section 2.3.

Incorporating a measurement $\mathbf{z}_j$, where $j \notin U$, into the current belief state $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U})$ is accomplished by further conditioning the belief with the new measurement. Hence, the new belief state is

$$p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\}) \ .$$

From estimation theory, this distribution can be computed with knowledge of $\mathbf{h}$, λ_i, and $\mathbf{z}_j$ and some appropriate independence assumptions. Incorporating a measurement $\mathbf{z}_j$ has the effect of mapping an element of $\mathcal{P}(\mathcal{R}^d)$ to another element of $\mathcal{P}(\mathcal{R}^d)$. Since ψ gives a measure of how "tight" a distribution in $\mathcal{P}(\mathcal{R}^d)$ is, it is clear that the best sensor $\hat{j} \in A = \{1, \ldots, N\} - U$ to choose is

$$\hat{j} = \arg_{j \in A} \max \ \psi \left(p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\}) \right) \ .$$

However, in practice, we only have knowledge of $\mathbf{h}$ and λ_i to determine which sensor to choose. In other words, we don't know the measurement value $\mathbf{z}_j$ before it is being sent. Nevertheless, we wish to select the "most likely" best sensor. Hence, it is necessary to marginalize out the particular value of $\mathbf{z}_j$. Note that for any given value of $\mathbf{z}_j$ for sensor j , we get a particular value for $\psi(\cdot)$ acting on the new belief state $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\})$. Now, for each sensor j, consider the set of all values of $\psi(\cdot)$ for choices of $\mathbf{z}_j$. Here are some possibilities for summarizing these set of values of $\psi(\cdot)$ by a single quantity:

Best average case:

$\hat{j} = \arg_{j \in A} \max E_{\mathbf{z}_j} [\psi(p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\})) \mid \{\mathbf{z}_i\}_{i \in U}]$

For a particular j, $E_{\mathbf{z}_j}[\cdot]$ represents the average information utility over the set of new belief states weighted by $p(\mathbf{z}_j \mid \{\mathbf{z}_i\}_{i \in U})$. The sensor $\hat{j}$ chosen is the one with best average information utility.

Maximizing worst case:

$$\hat{j} = \arg_{j \in A} \max \min_{\mathbf{z}_j} \psi(p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\}))$$

In this case, for a particular j, we take the pessimistic view that nature will present us with a measurement value $\mathbf{z}_j$ that would give us the worst possible information utility. Hence, we are maximizing our worst suspicions.

Maximizing best case:

$$\hat{j} = \arg_{j \in A} \max \max_{\mathbf{z}_j} \psi(p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\}))$$

In this case, for a particular j, we take the optimistic view that nature will provide us with a measurement value $\mathbf{z}_j$ that would give us the best possible information utility. Hence, we are maximizing our most optimistic beliefs.

Section 2.3 will present some possibilities of the particular form of the information utility $\psi(\cdot)$.

2.3 Information Utility Measures

In order to quantify the information gain provided by a sensor measurement, it is necessary to define a measure of information utility. The intuition we would like to exploit is that information content is inversely related to the "size" of the high probability uncertainty region of the estimate of $\mathbf{x}$. This section will define several such measures.

Covariance-Based

In the simplest case of a uni-modal posterior distribution that can be approximated by a Gaussian, we can derive utility measures based on the covariance Σ of the distribution $p_X(x)$. The determinant $\det(\Sigma)$ is proportional to the volume of the rectangular region enclosing the covariance ellipsoid. Hence, the information utility function for this approximation can be chosen as:

$$\psi(p_X) = -\det(\Sigma) \ .$$

Although the volume of the high probability region seems to be a useful measure, there are cases in which this measure is underestimating the residual uncertainty. In case the smallest principal axis shrinks to zero, the volume of the uncertainty ellipsoid is zero, while the uncertainties along the remaining principal axes might remain large.

An alternative measure using only the covariance Σ of a distribution $p_X(x)$ would be the trace $\mathrm{trace}(\Sigma)$, which is proportional to the circumference of the rectangular region enclosing the covariance ellipsoid. Hence, the information utility function would be

$$\psi(p_X) = -\mathrm{trace}(\Sigma) \ .$$

Fischer Information Matrix

An alternative measure of information is the Fisher information matrix $\mathbf{F}(\mathbf{x})$ defined over a class of likelihood densities $\{p(z_1^N \mid \mathbf{x})\}_{\mathbf{x} \in S}$, where z_1^N refers to the sequence $z_1, \ldots, z_N$ and x takes values from space S. The ij^{th} component of $\mathbf{F}(\mathbf{x})$ is

$$\mathbf{F}_{ij}(\mathbf{x}) = \int p(z_1^N \mid \mathbf{x}) \frac{\partial}{\partial x_i} \ln p(z_1^N \mid \mathbf{x}) \frac{\partial}{\partial x_j} \ln p(z_1^N \mid \mathbf{x}) dz_1^N$$

where x_i is the i^{th} component of $\mathbf{x}$. The Cramer-Rao bound states that the error covariance Σ of any unbiased estimator of $\mathbf{x}$ satisfies

$$\Sigma \geq \mathbf{F}^{-1}(\mathbf{x}) \ .$$

It can be shown that Fisher information is related to the surface area of the high probability region which is a notion of the "size" of the region [Cover&Thomas 1991]. Similar to the covariance-based measures, possible forms of the information utility function using the Fisher information are

$$\psi(p_X) = \det(\mathbf{F}(\mathbf{x})),$$

quantifying the inverse of the volume of high probability uncertainty region, or

$$\psi(p_X) = \mathrm{trace}(\mathbf{F}(\mathbf{x})) \ .$$

However, calculation of the Fisher information matrix requires explicit knowledge of the distribution. For the case when a Gaussian distribution can approximate the posterior, the Fisher information matrix is the inverse of the error covariance:

$$\mathbf{F} = \Sigma^{-1} \ .$$

If additionally the Markov assumption for consecutive estimation steps holds, we can incrementally update the parameter estimate using a Kalman filter for linear models. In this case, the Fisher information matrix can be updated recursively and independent of measurement values using the Kalman equations [Mutambara 1998]:

$$\mathbf{F}^{(k)} = \mathbf{F}^{(k-1)} + (\mathbf{H}^{(k)})^T (\mathbf{R}^{(k)})^{-1} \mathbf{H}^{(k)}, \qquad (5)$$

where $\mathbf{H}^{(k)}$ and $\mathbf{R}^{(k)}$, are the observation matrix, (2), and the measurement noise covariance at estimation step k, respectively.

For nonlinear systems, a popular approach is to use the extended Kalman Filter, which is a linear estimator for nonlinear systems, obtained by linearization of the nonlinear state and observation equations. In this case, the information matrix $\mathbf{F}$ can be recursively updated by [Mutambara 1998]:

$$\mathbf{F}^{(k)} = \mathbf{F}^{(k-1)} + (\mathbf{J}^{(k)})^T (\mathbf{R}^{(k)})^{-1} \mathbf{J}^{(k)} \qquad (6)$$

where $\mathbf{J}$ is the Jacobian of the measurement model $\mathbf{h}(\cdot)$ in (1). The information gain can then be measured by the "size" of this information matrix such as the determinant or trace.

Entropy of Estimation Uncertainty

If the distribution of the estimate is highly non-Gaussian (e.g., multi-modal), then the covariance Σ is a poor statistic of the uncertainty. In this case, one possible utility measure is the information-theoretic notion of information: the entropy of a random variable. For a discrete random variable X taking values in a finite set S, the Shannon entropy is defined to be

$$H(X) = -\sum_{x \in S} P(X = x) \log P(X = x) \ .$$

For a continuous random variable X taking values in a continuous space S , the equivalent notion is the differential entropy

$$h(X) = -\int_S p_X(x) \log p_X(x) dx \ .$$

The entropy can be interpreted as the log of the volume of the set of typical values for random variable X [Cover&Thomas 1991]. While this measure relates to the volume of the high probability uncertainty region, the computation of the entropy requires knowledge of the distribution $p_X(x)$ and can be computationally intensive for a general $p_X(x)$. Note that entropy is a function of the distribution only and not a function of the particular values that random variable X takes. Furthermore, entropy is a measure of uncertainty which is inversely proportional to our notion of information utility, and we can define the information utility as

$$\psi(p_X) = -H(P)$$

for discrete random variable X with probability mass function P and

$$\psi(p_X) = -h(p_X)$$

for a continuous random variable X with probability density p_X. Note that the notation for the entropy has been changed above from the original definition to reflect the fact that entropy is a function of distribution only.

Volume of High Probability Region

Another measure for non-Gaussian distributions could be the volume of the high probability region Γ_β of probability $\gamma \in (0, 1]$ defined to be

$$\Gamma_\beta = \{\mathbf{x} \in S : p(\mathbf{x}) \geq \beta\}$$

where β is chosen so that $P(\Gamma_\beta) = \gamma$. P is a probability measure on S, and $p(\cdot)$ represents the corresponding density. The relation between this measure of the volume and entropy is that entropy does not arbitrarily pick a value of γ [Cover&Thomas 1991]. The information utility function for this approximation is

$$\psi(p_X) = -\mathrm{vol}(\Gamma_\beta) \ .$$

Sensor Geometry Based Measures

In some cases, the utility of a sensor measurement is a function of the geometric location of the sensors only. For example, if the signal model assumes the form (4), the contribution of each individual sensor in terms of the likelihood function is an annulus centered about the position of the sensor. If, at a given time step, the current estimate of the target location is a Gaussian with mean $\hat{\mathbf{x}}$ and covariance Σ, then intuitively, the sensor along the longest principal axis of the covariance ellipsoid provides better discrimination on the location of the target than those located along the shortest principal axis of the uncertainty ellipsoid. This is due to the fact that each amplitude measurement provides a distance constraint (in the shape of the annulus) on the location of the target, and incorporating that constraint amounts to intersecting the annulus with the ellipsoid. Hence, we would get less uncertainty about the position of the target if the major axis of the ellipsoid were perpendicular to the tangent of the annulus. Furthermore, it is desirable that the sensor be closer to the mean of the current belief which affects the thickness of the annulus due to the additive nature of the noise term in (4).

In this special case, the geometric measure

$$\psi(p_X) = (\mathbf{x}_j - \hat{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x}_j - \hat{\mathbf{x}})$$

based on the Mahalanobis distance of the sensor under consideration to the current position estimate is an appropriate utility measure. The Mahalanobis distance measures the distance to the center of the error ellipsoid, normalized by the covariance Σ. By choosing the sensor j with the smallest Mahalanobis distance, we are incorporating the amplitude measurement which we believe will provide the most reduction in the uncertainty of the current belief.

Certainly, for other types of sensors, choosing the sensor based on minimizing Mahalanobis distance is not appropriate. For example, for sensors measuring bearing (e.g., by beamforming), the best sensor to choose is the sensor along the shortest axis of the error ellipsoid when the bearing sensor is pointing towards the ellipsoid.

2.4 Composite Objective Function

Up till now, we have ignored the communication cost of transmitting information across the network, and we have also ignored which sensor actually holds the current belief. In the remainder of this paper we will refer to the sensor, l, which holds the current belief, as the leader node. This node might act as a relay station to the user, in which case the belief resides at this node for an extended time interval, and all information has to travel to this leader. In another scenario, the belief itself travels through the network, and nodes are dynamically assigned as leaders. Depending on the network architecture and the measurement task, both or a mixture of both cases can be implemented. Without loss of generality, we assume that the leader node temporarily holds the belief state and that information has to travel a certain distance through the network to be incorporated into the belief state.

In order to incorporate the actual cost of information transport, the objective function for sensor querying and routing is a function of both information utility and cost of bandwidth and latency. This can be expressed by a composite objective function, M_c, of the form

$$\begin{aligned} M_c\left(\lambda_l, \lambda_j, p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i\in U})\right) &= \gamma M_u\left(p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i\in U}), \lambda_j\right) \\ &- (1-\gamma)\, M_a\left(\lambda_l, \lambda_j\right), \end{aligned} \quad (7)$$

where $M_u\left(p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i\in U}), \lambda_j\right)$ is one of the information utility functions ψ defined in Section 2.2, and M_a measures the cost of the bandwidth, and latency of communicating information between sensor j and sensor l. The tradeoff parameter $\gamma \in [0, 1]$ balances the contribution from the two terms in (7). The objective is to maximize M_c by selecting a sensor j from the remaining sensors $A = \{1, \ldots, N\} - U$ by

$$\hat{j} = \arg_{j\in A} \max M_c\left(\lambda_l, \lambda_j, p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i\in U})\right)$$

To illustrate this by an example, let us assume that the uncertainty in the position estimate can be approximated by a Gaussian probability distribution about the estimated position $\hat{\mathbf{x}}$, i.e., $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i\in U}) = N(\hat{\mathbf{x}}, \Sigma)$. As outlined above, an appropriate information utility function at the sensor position $\lambda_j = \mathbf{x}_j$ is given by the Mahalanobis distance for amplitude measuring sensors. If we impose an additional constraint on the distance between the querying sensor (l) and the queried sensor (j), the composite objective function comprises the following two terms:

$$M_u(\mathbf{x}_j, \hat{\mathbf{x}}, \Sigma) = -(\mathbf{x}_j - \hat{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x}_j - \hat{\mathbf{x}}) \quad (8)$$

$$M_a(\mathbf{x}_j, \mathbf{x}_l) \quad = \quad (\mathbf{x}_j - \mathbf{x}_l)^T (\mathbf{x}_j - \mathbf{x}_l) \qquad (9)$$

where $\mathbf{x}_l$ represents the position of the querying sensor l.

2.5 Representation and Incremental Update of Belief State

Recall that the belief is defined as the posterior probability of the random variable $\mathbf{x} \in \mathcal{R}^d$ given the measurements $\mathbf{z}_1$, $\mathbf{z}_2$, ..., $\mathbf{z}_n$. Assuming densities exist, this is $p(\mathbf{x}|\mathbf{z}_1, ..., \mathbf{z}_n)$.

Let us assume that *a priori*, $\mathbf{x}$ is uniformly distributed over some compact subset $S \subset \mathcal{R}^d$, and $\mathbf{z}_i$ are conditionally independent with respect to $\mathbf{x}$. Then, the likelihood can be factored as

$$p(\mathbf{z}_1, ..., \mathbf{z}_n|\mathbf{x}) = \prod_i p(\mathbf{z}_i|\mathbf{x})$$

And the belief is equivalent to the likelihood up to some multiplicative constant c_1:

$$\begin{aligned} p(\mathbf{x}|\mathbf{z}_1, ..., \mathbf{z}_n) &= c_0 p(\mathbf{z}_1, ..., \mathbf{z}_n|\mathbf{x}) p(\mathbf{x}) \\ &= c_1 p(\mathbf{z}_1, ..., \mathbf{z}_n|\mathbf{x}) \\ &= c_1 \prod_i p(\mathbf{z}_i|\mathbf{x}) \end{aligned}$$

where c_0 is a normalizing constant and the second equality follows from the fact that $\mathbf{x}$ is uniformly distributed in S. Due to this equivalence by the assumptions above, we can use the term belief and likelihood interchangeably. Since it is difficult to represent the exact belief when the belief is some arbitrary density, it is necessary to approximate the belief. Since the belief needs to be communicated to remote sensors in certain scenarios, a compact representation is desirable. Details of the tradeoffs between parametric and nonparametric representations will be postponed until Section 5.

Another desirable characteristic of the representation of the belief is if the representation can be incrementally updated. Under the assumptions above, the belief

$$p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_n)$$

can be computed from the previous belief $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_{n-1})$ and the new measurement $\mathbf{z}_n$ by

$$p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_n) = c_2 p(\mathbf{z}_n \mid \mathbf{x}) p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_{n-1})$$

where c_2 is a normalizing constant.

For the special case of a linear system with Gaussian disturbances, the well-known Kalman filter is an algorithm which incrementally computes the mean and covariance of the estimate. Since it is known that the estimate is Gaussian, the filter computes the optimal estimates. For nonlinear systems, the extended Kalman filter has been successfully applied. Another incremental algorithm is the sequential Monte Carlo method (also known as particle filter or condensation algorithm) [Doucet et al. 2000]. Without getting into the details of these incremental techniques, the point is that such incremental update mechanisms are desirable in order to easily fuse new measurements with the current belief. Furthermore, in order to pass the belief around easily, it is desirable that the representation of the belief be a compact one.

3 Algorithms

We describe two classes of algorithms, one based on the fixed belief carrier protocol in which a designated node such as a cluster leader holds the belief state, and another based on the dynamic belief carrier protocol in which the belief is successively handed off to sensor nodes closest to locations where "useful" sensor data are being generated. In the first case, the querying node selects optimal sensors to request data from using the information utility measures. For example, using the Mahalanobis distance measure, the querying node can determine which node can provide the most useful information while balancing the energy cost, without the need to have the sensor data first. In the dynamic case, the current sensor node updates the belief with its measurement and sends the estimation to the next neighbor that it determines can best improve the estimation. Although the algorithms presented here assume there is a single belief carrier node active at a time, the basic ideas also apply to scenarios where multiple belief carriers can be active simultaneously.

3.1 Information-Driven Sensor Query

This section outlines a sensor selection algorithm based on the cluster leader type of distributed processing protocol. Although the algorithm is presented using the cluster leader protocol, the ideas of guiding sensor selection using an information-driven criterion can be equally well supported by other methods such as the directed diffusion routing. The description of the algorithm below will explicitly discuss what information each sensor node has, even though this part of the algorithm is auxiliary to the sensor selection aspect of the algorithm. Assume we have a cluster of N sensors each labelled by a unique integer in $\{1, \ldots, N\}$. *A priori*, each sensor i only has knowledge of its own position $\mathbf{x}_i \in \mathcal{R}^2$. Figure 2 shows the flowchart of this algorithm which is identical for every sensor in the cluster. The algorithm works as follows:

1 Initialization Assuming all sensors are synchronized so that they are running the initialization routine at the same time, the first computation is to pick a leader from the cluster of N sensors. Depending on how the leader is determined, the sensors will have to communicate information about their position. For example, [Gao et al. 2001] describes a clustering algorithm with mobile centers. Leaving out the details of this leader selection, let us assume the sensor node labelled l is the leader. Assume also that the leader node has knowledge of certain characteristics $\{\lambda_i\}_{i=1}^N$ of the sensors in the network such as the positions of the sensor nodes.

2a Follower Nodes If the sensor node is not the leader, then the algorithm follows the left branch in Figure 2. These nodes will wait for the leader node to query them, and if they are queried, they will process their measurements and transmit the queried information back to the leader.

2b Initial Sensor Reading If the sensor node is the leader, then the algorithm follows the right branch in Figure 2. When a target is present in the range of the sensor cluster, the cluster leader will become activated (e.g. the amplitude reading at the leader node is greater than some threshold). The leader node will then

1. calculate a representation of the belief state with its own measurement, $p(\mathbf{x} \mid \mathbf{z}_l)$, and
2. begin to keep track of which sensors' measurements have been incorporated into the belief state, $U = \{l\} \subset \{1, \ldots, N\}$.

Again, it is assumed that the leader node has knowledge of the characteristics $\{\lambda_i\}_{i=1}^N$ of all the sensors within the cluster.

3 Belief Quality Test If the belief is good enough, based on some measure of goodness, the leader node is finished processing. Otherwise, it will continue with sensor selection.

4 Sensor Selection Based on the belief state, $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U})$, and sensor characteristics, $\{\lambda_i\}_{i=1}^N$, pick a sensor node from $\{1, \ldots, N\} - U$ which satisfies some information criterion $\psi(\cdot)$. Say that node is j. Then, the leader will send a request for sensor j's measurement, and when the leader receives the requested information, it will

1. update the belief state with $\mathbf{z}_j$ to get a representation of
$$p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \mathbf{z}_j) \ , \text{ and}$$
2. add j to the set of sensors whose measurements have already been incorporated
$$U := U \cup \{j\} \ .$$

Now, go back to step 3 until the belief state is good enough.

At the end of this algorithm, the leader node contains all the information about the belief from the sensor nodes by intelligently querying a subset of the nodes which provide the majority of the information. This reduces unnecessary power consumption by transmitting only the most useful information to the leader node. This computation can be thought of as a local computation for this cluster. The belief stored by the leader can then be passed up for processing at higher levels.

Note that this algorithm contains the general ideas discussed earlier in this paper. In steps 2b and 4, some representation of the belief $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U})$ is stored at the leader node. Considerations for the particular representation of the belief was mentioned in Section 2.5 and is discussed in detail in Section 5. In step 4, an information criterion is used to select the next sensor. Different measures of information utility were discussed in Section 2.3, each with their own computational complexity issues and notions of information content.

3.2 Constrained Anisotropic Diffusion Routing

While the Information-Driven Sensor Query algorithm provides us with a way of selecting the optimal order of sensors to provide maximum incremental information gain, it does not specifically define how both, the query and the information, are routed between the querying and the queried sensor. This section outlines a number of algorithms that exploit the composite objective function to dynamically determine the optimal routing path.

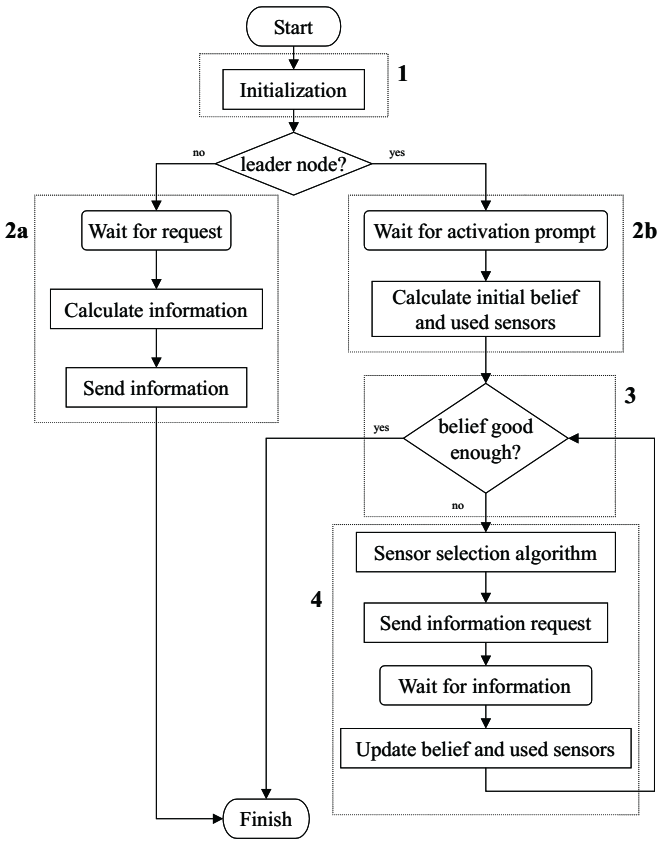

Figure 2: Flowchart of the information-driven sensor querying algorithm for each sensor.

Global knowledge of sensor positions
If the querying sensor has global knowledge of all sensor positions in the network, the routing can be directly addressed to the sensor closest to the optimal position. The optimal position $\mathbf{x}_o$ can be computed by the querying sensor by evaluating the composite objective function, M_c:

$$\mathbf{x}_o = \arg_{\mathbf{x}} \left[\nabla M_c = 0 \right]. \tag{10}$$

If the underlying routing protocol layer supports absolute sensor addresses, the querying sensor can directly request information from the sensor located closest to the optimum position.

Routing without global knowledge of sensor positions
Here the information query is directed by local decisions of individual sensor nodes and guided into regions satisfying the constraints defined by M_c. Note that M_c is incrementally updated as the belief is updated along the routing path. The local decisions can be based upon different criteria:

1. For each sensor located at $\mathbf{x}_k$, evaluate the objective function M_c at the positions of the m sensors within a local neighborhood determined by the communication distance, and pick the sensor j that maximizes the objective function locally within the neighborhood:
$$\hat{j} = \arg_j \max(M_c(\mathbf{x}_j)), \ \forall j \neq k$$

where $\mathbf{x}_k$ is the position of the current routing node.

2. Choose the next routing sensor in the direction of the gradient of the objective function, ∇M_c. Among all sensors within the local communication neighborhood, choose the sensor j such that

$$\hat{j} = \arg_j \max \left(\frac{(\nabla M_c)^T (\mathbf{x}_k - \mathbf{x}_j)}{|\nabla M_c| \, |\mathbf{x}_k - \mathbf{x}_j|} \right),$$

where $\mathbf{x}_k$ denotes the position of the current routing node.

3. Instead of following the local gradients of the objective function throughout the routing path, the chosen direction at any hop can be biased towards the direction aiming at the optimum position, $\mathbf{x}_o$. This variation of the gradient ascent algorithm is most useful in regions of small gradients of the objective function, i.e., where the objective function is flat. The direction towards the minimum of the objective function can be found by evaluating (10) at any routing step. This allows to locally computing the direction towards the optimum position, i.e., $(\mathbf{x}_o - \mathbf{x}_k)$, where $\mathbf{x}_k$ denotes the position of the current routing sensor. The optimal direction towards the next sensor can be chosen according to a weighted average of the gradient of the objective function and the direct connection between the current sensor and the optimum position:

$$\mathbf{d} = \beta \nabla M_c + (1 - \beta)(\mathbf{x}_o - \mathbf{x}_j),$$

where the parameter β can be chosen, for example, as a function of the distance between the current and the optimum sensor position: $\beta = \beta(|\mathbf{x}_o - \mathbf{x}_j|)$. This routing mechanism allows adapting the routing direction to the distance from the optimum position. For large distances it might be better to follow the gradient of the objective function for the steepest ascent, i.e., the fastest information gain. For small distances about the optimum position, the objective function is flat and it is faster to directly go towards the minimum than following the gradient ascent.

In order to locally evaluate the objective function and its derivatives, the query needs to be transmitted together with information on the current belief state. This information should be a compact representation of the current estimate and its uncertainty, and has to provide complete information to incrementally update the belief state given local sensor measurements. For the above example of quantifying the information utility by the Mahalanobis distance, we need to transmit the triplet $\{\hat{\mathbf{x}}, \mathbf{x}_q, \hat{\Sigma}\}$ with the query, where $\hat{\mathbf{x}}$ is the current state of the estimated target position, $\mathbf{x}_q$ is the position of the querying sensor, and $\hat{\Sigma}$ is the current estimate of the position uncertainty covariance.

In-network processing

The above-mentioned routing mechanisms can be used to establish a routing path towards the potentially best sensor along which the measurement from the sensor closest to the optimal position is shipped back. In the case of global knowledge of sensor positions, the routing path is optimal. In the case of local sensor knowledge, the path is only locally optimal as the routing algorithm is a greedy algorithm. The estimate and the estimation uncertainty can be dynamically updated along the routing path. The measurement can also be shipped back to the query originating node. Since the information utility objective function along the path is monotonically increasing, the information provided by subsequent sensors is getting incrementally better towards the global optimum. When the information is continuously shipped back to the querying sensor, the information arriving in sequential order provides an incremental improvement to the estimate. Once predefined estimation accuracy is reached, the querying sensor can terminate the query even if it has not yet reached the optimal sensor position. Alternatively, instead of shipping information back to the querying sensor, the result could be read out from the network at the sensor where the information resides.

Incorporate target dynamics into query

For moving targets, during the local dynamic update of the believe state and the objective function, a model on the target dynamics can be used to predict the position of the target in the next time step. This predicted target position and the associated uncertainty can be used to dynamically aim the information directed query at future positions to optimally track the target.

4 Experimental Results

This section describes two sets of experiments aimed at validating the IDSQ and CARD algorithms presented earlier.

4.1 Information-Driven Sensor Querying (IDSQ)

We will apply the sensor selection algorithm presented in the Section 3.1 to the problem of spatial localization of a stationary target based on amplitude measurements from a network of sensors.

The measurement model for each sensor i is

$$z_i = \frac{a}{||\mathbf{x} - \mathbf{x}_i||^{\frac{\alpha}{2}}} + w_i \tag{11}$$

for $i = 1, \ldots, N$ where

- $a \in \mathcal{R}$ is the amplitude of the target uniformly distributed in the interval $[a_{low}, a_{high}]$,
- $\mathbf{x} \in \mathcal{R}^2$ is the unknown target position,
- $\mathbf{x}_i \in \mathcal{R}^2$ is the known sensor position,
- $\alpha \in \mathcal{R}$ is the known attenuation coefficient, and
- w_i is white, zero-mean Gaussian noise with variance σ_i^2.

The purpose of these experiments is to compare different sensor selection criteria. Issues of compactly representing the belief state will be postponed to Section 5.3. The representation θ of the belief will be the history of the collected measurements from the sensors. Thus, the true belief $p(\mathbf{x} \mid \theta)$ and any statistics thereof can be calculated.

Let us fill in the details of the algorithm presented in Section 3.1.

1 Initialization The relevant characteristics of each sensor i are

$$\lambda_i = \begin{bmatrix} \mathbf{x}_i \\ \sigma_i^2 \end{bmatrix}$$

where $\mathbf{x}_i$ is the position and σ_i^2 is the variance of the additive noise term. For simplicity, the leader is chosen to be the one whose position $\mathbf{x}_i$ is closest to the centroid of the sensors, that is,

$$l = \arg_{j=1,\ldots,N} \min \|\mathbf{x}_j - \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\| \ .$$

To determine the leader, all sensors communicate their characteristics λ_i to each other.

2a Follower Nodes When a follower node i is queried by the leader, it will transmit z_i to the leader.

2b Initial Sensor Reading For the experiments, the leader node l was activated when its amplitude reading satisfied

$$z_l > \gamma \ .$$

This essentially means the leader node becomes activated when the target is less than some given distance away from the leader node, assuming there is no other sound source present. The leader will then

1. store its amplitude value $\theta = z_l$, which is its representation of the belief, and
2. keep track of which sensors' measurements have been incorporated into the belief state,

$$U = \{l\} \ .$$

3 Belief Quality Test In the general case, a test to determine when the quality of the belief is good enough is needed here; however, for the purposes of the experiments, we will continue to incorporate measurements until all sensors in the cluster have been incorporated. All clusters in the experiments consist of 7 sensors.

4 Sensor Selection The information which is utilized to compute the next sensor to query is the belief state θ and sensor characteristics $\{\lambda_i\}_{i=1}^N$. We have four different criteria for choosing the next sensor $\hat{j}$:

A *Nearest Neighbor Data Diffusion*

$$\hat{j} = \arg_{j \in \{1,\ldots,N\}-U} \min \|\mathbf{x}_l - \mathbf{x}_j\|$$

B *Mahalanobis distance* First, calculate the mean and covariance of the belief state:

$$\mu = \int \mathbf{x} p(\mathbf{x} \mid \theta) d\mathbf{x}$$

$$\Sigma = \int (\mathbf{x}-\mu)(\mathbf{x}-\mu)^T p(\mathbf{x} \mid \theta) d\mathbf{x}$$

and choose by

$$\hat{j} = \arg_{j \in \{1,\ldots,N\}-U} \min (\mathbf{x}_j - \mu)^T \Sigma^{-1} (\mathbf{x}_j - \mu) \ .$$

C *Maximum likelihood* This is an ad hoc generalization of the Mahalanobis distance criterion for distributions that are multi-modal. For the special case when the true distribution is Gaussian, this criterion corresponds exactly with the Mahalanobis distance criterion.

$$\hat{j} = \arg_{j \in \{1,\ldots,N\}-U} \max p(\mathbf{x}_j \mid \theta) \ .$$

D *Best Feasible Region* This is an uncomputable method since this requires knowledge of the sensor value in order to determine the sensor to use. However, this has been implemented to serve as a basis for comparison of the other criteria.

$$\hat{j} = \arg_{j \in \{1,\ldots,N\}-U} \min \int_{\Gamma_\beta} d\mathbf{x}$$

where

$$\begin{aligned} \Gamma_\beta = \{\mathbf{x} \in \mathcal{R}^2 : \ & \exists \{w_i = n_i\}_{i \in U} \text{ with} \\ & \sqrt{\textstyle\sum_{i \in U} \frac{n_i^2}{\sigma_i^2}} \leq \beta \text{ s.t.} \\ & \forall i \in U \ \exists a \in [a_{low}, a_{high}] \text{ with} \\ & z_i = \frac{a}{\|\mathbf{x}-\mathbf{x}_i\|^{\frac{\alpha}{2}}} + n_i \} \end{aligned}$$

Viewing the w_i's as a vector of independent normal variables with standard deviation σ_i, β is the standard deviation of this multivariate random variable. β then controls the maximum energy of the noise instances. The set of $\mathbf{x} \in \Gamma_\beta$ is the set of target positions which could have generated the measurements $\{z_i\}_{i \in U}$

After node $\hat{j}$ has been determined, a request transmitted to node $\hat{j}$, and z_j received, the leader node will

1. update the representation of the belief state[1]

$$\theta := \theta \cdot z_{\hat{j}} \ , \text{ and}$$

2. update the set of used sensors

$$U := U \cup \{\hat{j}\} \ .$$

For the simulations, we have chosen $a_{low} = 10$ and $a_{high} = 50$. The sensor variance σ_i is set to 0.1, which is about 10% of the signal amplitude when the amplitude of the target is 30 and the target is at a distance of 30 units from the sensor. β is chosen to be 2 since this value covers 99% of all possible noise instances. For the first simulation, α was set to 1.6 which considers reflections from the ground surface. α is set to 2 for the second simulation which is the attenuation exponent in free space with no reflections or absorption. The shape of uncertainty region is sensitive to different choices of α; however, the comparative performance of the sensor selection algorithm for different selection criteria generally behave similarly for different α.

[1] Note that here we use the string concatenation operator as a shorthand to denote the belief update without getting into details of specific update algorithms. More discussion on this can be found later in Section5.

The first simulation is a pedagogical example to illustrate the usefulness of incorporating a sensor selection algorithm into the sensor network. Figure 3 shows the layout of 14 microphones, 13 of which are in a linear array and one which is perpendicularly off from the leader node of the linear array. The one microphone not in the linear array is placed so that it is farther from the leader node than the farthest microphone in the linear array. With sensor measurements generated by a stationary object in the middle of the sensor network, sensor selection criteria **A** and **B** are compared. The difference in the two runs is the order in which the sensors' measurements are incorporated into the belief.

Figure 4 shows a plot of the number of sensors incorporated versus the logarithm of the determinant of the error covariance of the belief state. Indeed, the volume of the error covariance under selection criterion **B** is less than the volume of the error covariance under selection criterion **A** for the same number of sensors, except during the initial phase or after all the sensors have been accounted for.

A plot of the amount of the communication distance versus the number of sensors incorporated is shown in Figure 5. Certainly, the curve for selection criterion **A** is the lower bound for any other criterion. **A** optimizes the network to use the minimum amount of communication energy when incorporating sensor information; however, it largely ignores the information content of these sensors. A more informative interpretation of the figure is to compare the amount of energy it takes for criterion **A** and criterion **B** to achieve the same level of accuracy. Examining Figure 6, we see that under criterion A, in order for the log determinant of the covariance value to be less than 5, criterion A requires all 14 sensors to be tasked. On the other hand, criterion B requires only 6 sensors to be tasked. Now, comparing the total communication distance for this level of accuracy from Figure 7, we see that criterion B requires less than 150 units of communication distance for tasking 6 sensors as opposed to nearly 500 units of communication distance for tasking all 14 sensors. Indeed, for a given level of accuracy, **B** generally requires less communication distance than **A**. With this comparison, criterion **B** performs better.

The above simulation was the run of a specific layout of the sensors, and the striking improvement of the error was largely due to the fact that most of the sensors were in a linear array. Thus, the next simulation will explore which one does better on the average with randomly placed sensors.

Microphones are placed uniformly in a given square region as shown in Figure 8 . The target is placed in the middle of the square region and given a random amplitude. Then, the sensor algorithm for the different sensor selection criteria described above was run for 200 runs. Figure 9 shows a comparison between selection criterion **A** and **B**. There are three segment in each bar. The bottom segment represents the percentage of runs when the error for **B** was strictly less than the error for **A** after k sensors have been incorporated. The middle represents a tie. The upper segment represents the percentage of runs when the error for **B** was larger than the error for **A**. Since the bottom segment is larger than the upper one (except for the initial and final cases), this shows **B** performs better than **A** on average.

Figure 10 and Figure 11 show comparisons of sensor criterion **C** and **D** versus **B**. The performance of **C** is comparable to **B**, and as expected, **D** is better than **B**. The reason **D** is not always better than **B** is because the nth best sensor is chosen incrementally with the first $n-1$ sensors fixed from before. Fixing the previous $n-1$ sensors when choosing the nth sensor is certainly suboptimal to choosing n sensors all at once to maximize the information content of the belief.

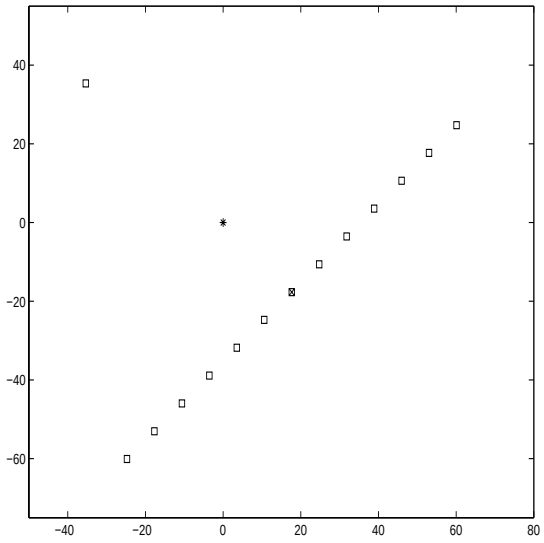

Figure 3: Layout of all-but-one-colinear sensors (squares) and target (asterisk). The leader node is denoted by an $\times$.

4.2 Constrained Anisotropic Diffusion Routing (CADR)

Figure 12 illustrates the idea of Constrained Anisotropic Diffusion Routing (CADR) by numerical simulations of ad hoc sensor networks of randomly placed sensors. The objective function was chosen according to (7), with the information utility and energy cost terms according to (8) and (9), respectively. The current target position, $\hat{\mathbf{x}}$, and its uncertainty, Σ, were arbitrarily chosen and remained fixed for the run, i.e., no incremental update of the belief state has been implemented. The value of the objective function across the sensor network is shown as a contour plot with peaks of the objective function located at the center of ellipses. The circle indicated by '?' depicts the position of the querying sensor (query origin), and the circle indicated by 'T' depicts the estimated target position, $\hat{\mathbf{x}}$. The current uncertainty in the position estimate, Σ, is depicted as 90-percentile ellipsoid enclosing the position 'T'.

The goal of the Constrained Anisotropic Diffusion Routing is to guide the query as close as possible towards the maximum of the objective function, following the local gradients to maximize incremental information gain. While the case of $\gamma = 1$ represents maximum information gain, ignoring the distance from the querying sensor (and hence the energy cost), the case $\gamma = 0$ minimizes the energy cost, ignoring the information gain. For other choices of $0 < \gamma < 1$, the composite objective function represents a tradeoff between information gain and energy cost.

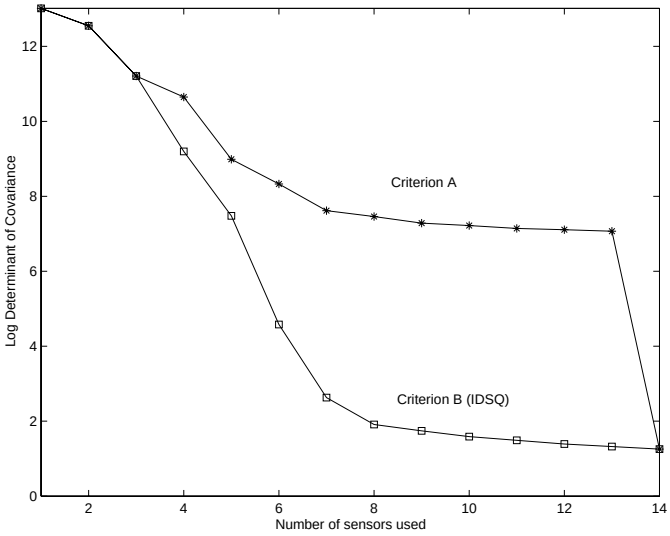

Figure 4: Determinant of the error covariance for selection criteria **A** and **B** (IDSQ) for the all-but-one-colinear sensor layout.

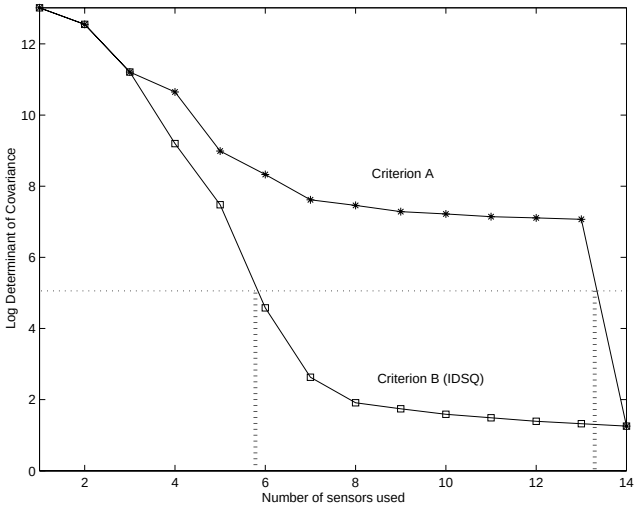

Figure 6: Determinant of the error covariance for selection criteria **A** and **B** (IDSQ) for the all-but-one-colinear sensor layout. **A** tasks 14 sensors while **B** tasks 6 sensors to be below an error threshold of 5 units.

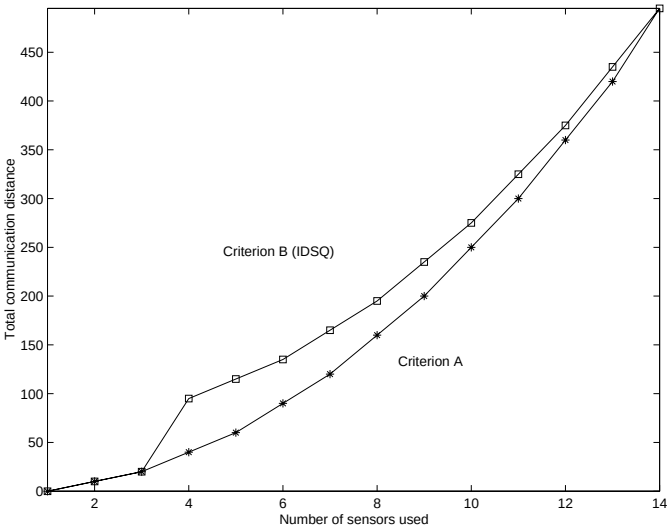

Figure 5: Total communication distance vs. the number of sensors queried for selection criteria **A** and **B** (IDSQ) for the all-but-one-colinear sensor layout.

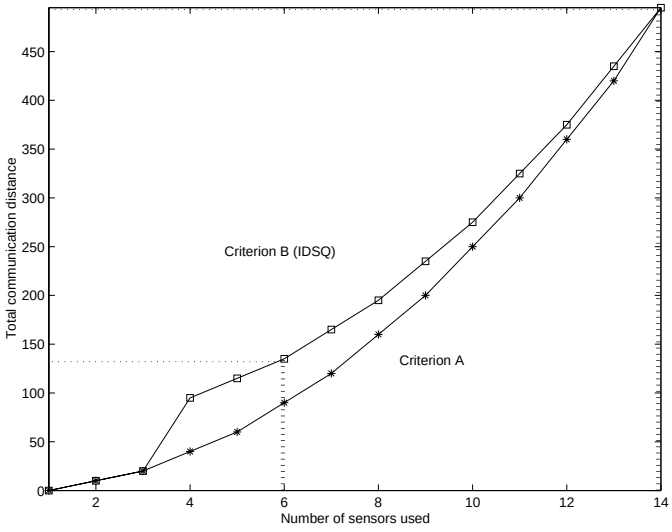

Figure 7: Total communication distance vs. the number of sensors queried for selection criteria **A** and **B** (IDSQ) for the all-but-one-colinear sensor layout. For achieving the same threshold of the error, **A** tasks 14 sensors and uses nearly 500 units of communication distance whereas **B** tasks 6 sensors and uses less than 150 units of communication distance.

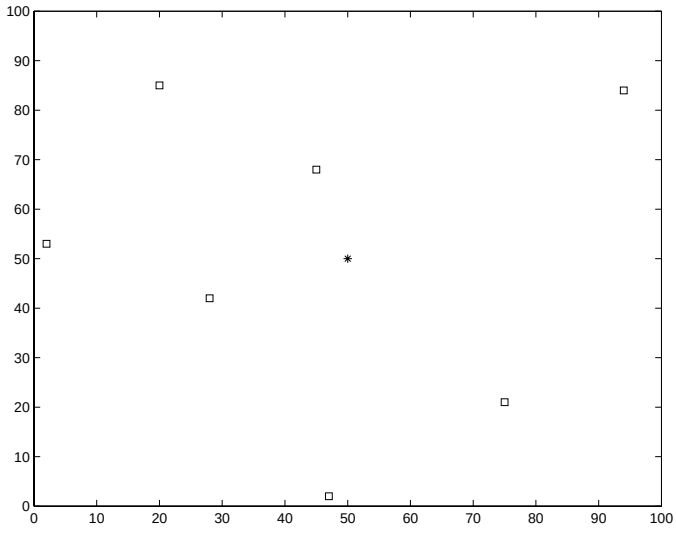

Figure 8: Layout of seven randomly placed sensors (squares) with target in the middle (asterisk).

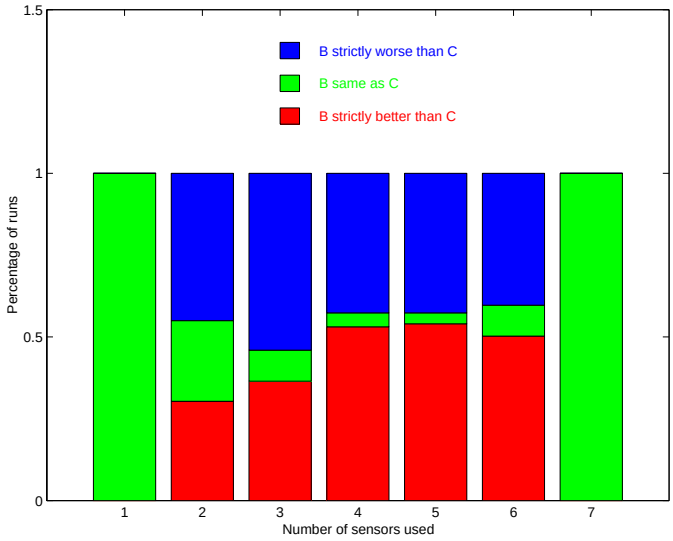

Figure 10: Percentage of runs where **B** performs better than **C** for seven randomly placed sensors.

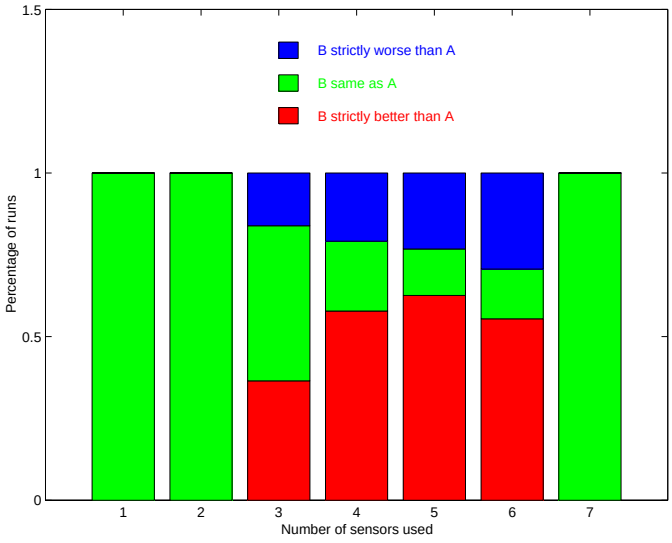

Figure 9: Percentage of runs where **B** performs better than **A** for seven randomly placed sensors.

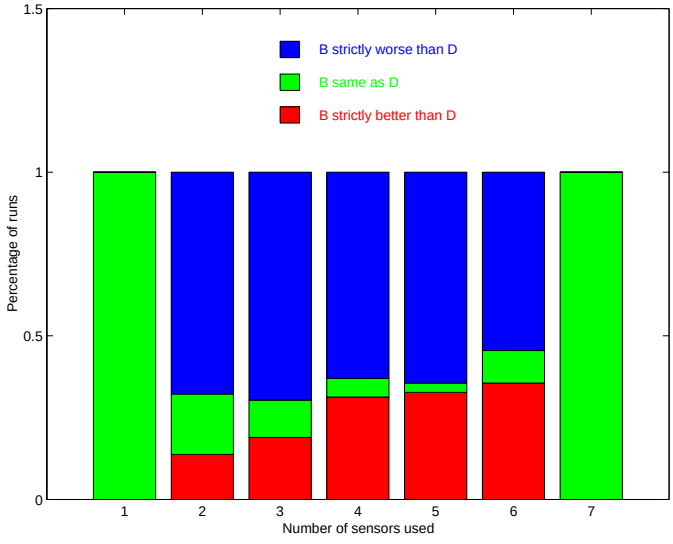

Figure 11: Percentage of runs where **B** performs better than **D** for seven randomly placed sensors.

Figure 12 shows how variation of the tradeoff parameter γ morphs the shape of the objective function. As γ decreases from 1 to 0, the peak location moves from being centered at the predicted target position ($\gamma = 1$) to the position of the querying sensor ($\gamma = 0$); at the same time, the contours change from being elongated, shaped according to the uncertainty ellipsoid represented by the estimated covariance Σ, towards isotropic. Another interesting aspect of the combined objective function is that the spatial position of its minimum does not shift linearly between the estimated target position 'T' and the query origin '?', with varying γ. This can be observed in the case of $\gamma = 0.2$, where the minimum is located off the line connecting 'T' and '?'.

In all three cases shown in Figure 12 the estimated target position and residual uncertainty are the same. Variations in shape and offset of the objective function are caused by variations of the tradeoff parameter γ. In order to visualize how the query is routed towards the maximum of the objective function by local decisions, both the estimated position $\hat{\mathbf{x}}$ as well as its uncertainty Σ, are left unaltered during the routing. It is important to note that incremental belief update during the routing by in-network processing would dynamically change both the shape and the offset of the objective function according to the updated values of $\hat{\mathbf{x}}$ and Σ at every node along the routing path. As the updated values of $\hat{\mathbf{x}}$ and Σ are passed on to the next node, all routing decisions are still made locally. Hence, the plotted objective function represents a snapshot of the objective function that an active routing node locally evaluates at a given time step, as opposed to the overlaid routing path which illustrates the temporal evolution of the multi-hop routing.

The routing in Figure 12 was terminated after reaching a spatial region where the value of the objective function is above a preset threshold. In real applications, the termination criterion can be chosen based upon other criteria, such as a threshold on the residual uncertainty or preset timeouts that are passed along with the query.

The small circles surrounding the dots along the routing path illustrate the subset of sensors the routing sensors (on the path) consider during sensor selection. Among these sensors the ones that locally maximize the objective function have been selected as the next routing nodes. The fraction of selected nodes among all nodes indicates the energy savings by using CADR, as opposed to the total energy cost of flooding the network.

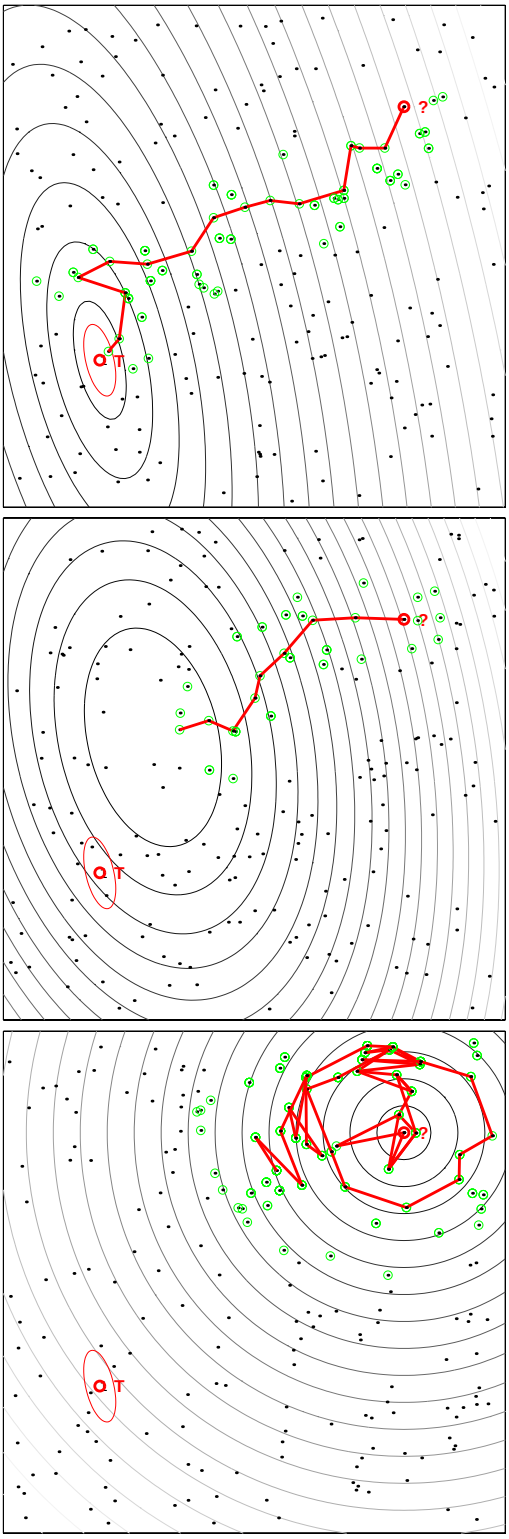

Figure 12: Constrained Anisotropic Diffusion Routing for $N = 200$ sensors, and varying information vs. cost tradeoff parameter γ. From top to bottom: $\gamma = 1$, $\gamma = 0.2$, $\gamma = 0.0$. For comparison, the position estimate of the target, 'T', and the position of the query origin, '?', are fixed in all examples.

5 Discussion

This section addresses several issues concerning practical implementations of the IDSQ/CADR algorithms.

5.1 Other Network Routing Support for IDSQ/CADR

We have described how CADR can effectively support intelligent sensor querying such as IDSQ. Existing data diffusion protocols such as [Intanagonwiwat et al. 2000] could be used to implement IDSQ/CADR. For simplicity, we will continue to use the localization of a single stationary target as an example.

As described earlier, in a cluster based network organization a user query enters into the network at the node called query proxy, often the cluster head itself. The proxy performs estimation of target states, using information from its sensors as well as sensors from the nodes in the cluster. If a target moves out of the range of the sensors in the cluster, then a hand-off needs to take place so as to get the current state estimation to the next cluster where the target is mostly likely going to be at.

Using the information utility measure such as the Mahalanobis distance described earlier, the query proxy selects a node in the cluster that can best improve the current belief state. Since the computation of Mahalanobis distance only requires the knowledge of the current belief state and the location of a sensor, this can be accomplished at the proxy node if the node has location information for all the sensors in the cluster. To request the data from the selected sensor, the proxy can perform a point-to-point communication using the address of the selected sensor node, or the proxy can turn on only the selected node (and when a node is turned on it sends measurement back to the proxy by default).

In the more general case, the node selection must be based on three pieces of information: belief state (e.g., mean, covariance for a Gaussian distribution), sensor location, and measurement at the sensor. We exploit the filter mechanism available in the directed diffusion routing protocol. Each node has a filter that acts on information it receives from a neighbor as well as its own measurement signal and decides to either send back the signal (traversing the information diffusion routes back to the proxy via callbacks), or change the data routing gradients it has already set up, or forward the information it receives to a chosen next node.

In one scenario, the query proxy broadcasts its current belief state to all nodes within its influence region. Some nodes may require more than a single hop to reach, using for example an underlying diffusion routing tree. There are two cases in which the sensor selection can be accomplished in-network: (1) Each sensor node that receives the broadcasted belief state computes a quality factor r indicating how well the current measurement can improve the current estimate using measures such as volume of error covariance, and those that meet certain threshold sends their data back. For example, every node that can reduce the estimate uncertainty by 50% or more sends data back to the proxy. (2) Each node sends a tuple $\{data, r\}$ back to the proxy, traversing the diffusion routing tree. At each junction of the tree where multiple messages collide, only the message with max r gets forwarded to the next node. This way the query proxy receives only the best measurement. Alternatively, in both (1) and (2), in addition to in-network sensor selection, the estimation can be accomplished in-network. For example, using the independent likelihood pool principle [Manyika&Durrant-Whyte] (assuming conditioned on target location and amplitude), then local estimates can be combined incrementally when signals flow back to the proxy. However, this has to balance with the amount of information to transmit since in some cases the encoding of the belief state might take more bits than the measurement itself (e.g. in the non-parametric form).

5.2 Belief Representation

There are several ways to approximate an arbitrary belief state:

1. Approximate by a family of distributions $M \subset \mathcal{P}(\mathcal{R}^d)$, which is parameterizable by a finite dimensional parameter $\theta \in \Theta$, where $\mathcal{P}(\mathcal{R}^d)$is the set of all probability distributions over $\mathcal{R}^d$. An example of M is the family of Gaussian distributions on $\mathcal{R}^d$ where the finite dimensional parameter space Θ includes the mean and covariance of the Gaussian distributions.
2. Approximate by point samples. This is a brute force method of approximating the density of a continuous random variable by a probability mass function with support in a finite number of points of S. Let $\tilde{S} \subset S$ be some finite set of points of S. Then, we can approximate the density by a PMF with mass at each point of $\tilde{S}$ proportional to the density at that point. Two examples of this approximation are discretizing the subset of S by a grid and the particle filter approximation of distributions. A variant of this point sample approximation is to partition S and assign a probability value to each region; this is a histogram type approach.

We should mention that the first approximation is referred to as a parametric approximation whereas the second approximation is referred to as a nonparametric approximation. Although the terms parametric and nonparametric may seem to refer to two completely different ways of approximation, we would like to point out that the distinction is not so clear. In the second case, if we consider the points of $\tilde{S}$ with their associated probability value as the set of all parameters Θ, then we can consider the second approximation a parametric approach as well. The distinction between a parametric approximation and a nonparametric one depends on what we mean by the term "parameter".

5.3 Impact of Choice of Representation

The representation of the belief will impact the amount of communication power consumed when sending the belief state to a remote sensor. If we must pass the belief to a remote sensor, we are faced with the following tradeoff:

1. Representing the true belief by a nonparametric approximation will result in a more accurate approximation of the belief at the cost of more transmitted bits.
2. Representing the true belief by a parametric approximation with relatively few parameters will result in a poor approximation of the belief but with the benefit that fewer bits need to be transmitted.

The above two tradeoffs are general statements. However, there is another factor that is underlying the tradeoffs above. That factor is background knowledge of the possible set of beliefs. Let us elaborate.

Parametric case (e.g. Gaussian approximation)
The reason we are able to transmit relatively few bits under the parametric case is that it is assumed that all sensors are aware of the parametric class of beliefs. Knowledge of this parametric class is the background knowledge which allows for the

small number of bits to be transmitted. In the case where we approximate the belief by a Gaussian, all sensors know that the belief is represented by the parametric class of Gaussian distributions. Hence, only the mean and covariance need to be transmitted. The Kalman filter equations are recursive update equations of the mean and covariance of the Gaussian distribution.

Nonparametric case (e.g. discretization)
For the nonparametric case, there is no constant-size parameterization of the belief in general. However, assuming that the model of the measurements is known, we can parameterize the belief by storing a history of all measurements. In this case, the parameter space is $\Theta = \mathcal{R}^* = \bigcup_{n=1}^{\infty} \mathcal{R}^n$, the set of all finite length strings of real numbers. If $\theta_m = z_1 z_2 \ldots z_m$ is the parameter for the likelihood function

$$p(z_1, \ldots, z_m \mid \mathbf{x}) \ ,$$

then the parameter $\theta_{m+1} = z_1 z_2 \ldots z_m z_{m+1}$ for

$$p(z_1, \ldots, z_{m+1} \mid \mathbf{x})$$

is updated from θ_m by

$$\theta_{m+1} = \theta_m \cdot z_{m+1}$$

where "$\cdot$" denotes string concatenation. This update equation is trivial and suffers from increasing dimensionality. But if we are given knowledge of a highly nonlinear model for the measurements of the unknown variable, it may be that collecting a history of the measurements is a more compact representation initially. To elaborate, a Gaussian approximation of the likelihood function

$$p(z_1, z_2 \mid \mathbf{x})$$

would require storing the mean and the covariance which for this particular case would require 2 real numbers for the mean and 3 real numbers for the covariance for a total of 5 real numbers. However, to store the exact likelihood, we only require 2 real numbers corresponding to z_1 and z_2, with the implicit background knowledge of the form of the likelihood function.

If, for example, it turns out that after some number of measurements, the true likelihood function becomes unimodal and, hence, can be approximated well by a Gaussian, this motivates a *hybrid* representation:

1. Initially, the belief is parameterized by a history of measurements.
2. Once the belief begins to look unimodal, we will approximate the belief by a Gaussian and, hence, the parameterization of the belief includes the mean and covariance.

Since a Gaussian approximation is poor initially, the cost of maintaining a history of measurements is justified. Once the Gaussian approximation becomes reasonable, we can convert to such a representation and control the parameter dimensionality.

6 Related Work

Researchers in robotics have developed a number of algorithms for quantifying and estimating uncertainties in sensing and estimation applications. In particular, [Nakamura 1992] uses a geometric characterization of uncertainty in terms of error ellipsoids and developed a Kalman-filter like incremental estimation algorithm, assuming Gaussian distributions, for minimizing the uncertainty ellipsoids for a robotic position estimation problem. [Manyika&Durrant-Whyte] and [Mutambara 1998] developed a general information filter formulation for the incremental estimation problem, similar to the Kalman filter. IDSQ differs from these in that IDSQ uses a set of information criteria to *select* which sensors to get data from and then incrementally combine the data. The belief update in IDSQ could use either an information filter or a more general non-Gaussian technique such as the sequential Monte Carlo method (also known as particle filter) [Doucet et al. 2000].

IDSQ is related to work on active vision or active testing in computer vision research. [Geman&Jedynak 1996] described a method for selecting tests using an entropy measure on the current set of hypotheses for a satellite image based road tracking problem. In their method, a test is selected if it minimizes the residual uncertainty about the location of the road, using an online implementation of a decision tree algorithm. While sharing the same philosophy of minimizing uncertainty in an estimation via active test selection, IDSQ models the uncertainty in a geometric sense so that active sensor selection can be guided using the spatial configuration of the sensors.

Distributed Kalman filter exploits the fact that measurements are incorporated into the estimate by a weighted linear combination of estimate and measurement [Mutambara 1998]. Hence, estimates based on different clusters of measurements can be combined linearly to create an estimate based on all the measurements. CADR works based on this incremental nature of incorporating measurements into the estimate, but unlike the distributed Kalman filter, CADR discriminates between useful measurements and redundant measurements to minimize unnecessary communication costs.

Wirelessly networked sensors often deploy a multihop RF communication strategy to conserve energy [Pottie&Kaiser 2000]. Recent work in data diffusion routing and more generally energy-aware communication attempts to minimize power consumption by exploiting network parameters such as topologies or node power levels. [Intanagonwiwat et al. 2000] describes a directed diffusion routing protocol that uses diffusion coupled with reinforcement to establish shortest paths between data sources and data sinks in a publish-subscribe framework. The diffusion proceeds in an isotropic fashion, reaching nearest neighbors first. In [Sohrabi 2000], sensors in a network self-organize based on signal-noise-ratios to perform cooperative processing tasks. Energy cost for data transmission is explicitly used to plan paths for sending data. Similarly, [Singh et al 1998] determines minimum metric paths that minimize energy per packet as well as weighing the energy consumption by energy reserve on a node. This way communication traffic can be steered away from low-energy nodes. By comparison, CADR uses *both* communication or energy cost (e.g. expressed as a function of the distance) *and* information utility to anisotropically diffuse data, generalizing the cost model used by the diffusion routing protocol. CADR allows estimation to either center at a query proxy node or dynamically migrate among nodes. In another

related work, [Brooks et al. 2001] uses predictions from tracking to invoke sensor nodes with interest in a particular direction or region. In contrast, IDSQ/CADR uses the more general information utility measure and cost function to locally select the best sensors to query or route data to.

IDSQ/CADR is closely related to [Byers&Nasser 2000] that uses a utility function to optimize node selection. In their formulation, the utility function mapping a set of nodes to real values is simply a step or sigmoid function, without explicit modeling of network spatial configuration. IDSQ/CADR, in contrast, models the uncertainty from a general information-theoretic formulation and derives specific forms of the utility functions for localization and tracking problems so that routing algorithms can exploit the spatial configuration of the network to optimize node selection.

7 Conclusion

This paper presents a new approach to distributed, collaborative signal processing in heterogeneous ad hoc sensor networks. The two key ideas that are outlined in this paper are information-driven sensor querying (IDSQ) and constrained anisotropic diffusion routing (CADR). Introducing an information utility measure allows to dynamically select the best subset of sensors among all possible sensors within the sensor network.

The sensor selection is based upon constraints on the information gain, subject to additional constraints on energy cost and inter-node communication distance. Defining an incremental algorithm for the estimation problem, and communicating a compact representation of the belief state together with the sensor query, allows to incrementally update the belief during the routing of the query. Furthermore, all routing decisions are based upon local computations in the routing nodes exploiting the propagated belief state.

We presented experimental results on simulated sensor networks for the task of localization of stationary targets given a sound amplitude based measurement model. The results show that the information-driven sensor queries proposed in this paper are more energy efficient, have lower latency, and provide distributed anytime algorithms to mitigate the risk of link/node failures.

Acknowledgment

This work is supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract number F30602-00-C-0139 through the Sensor Information Technology Program. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

References

[Brooks et al. 2001] R.R. Brooks, C. Griffin, and D.S. Friedlander, Self-organized distributed sensor network entity tracking, this issue.

[Byers&Nasser 2000] J. Byers and G. Nasser, Utility-Based Decision-Making in Wireless Sensor Networks (Extended Abstract), Proceedings of IEEE MobiHOC 2000, Boston, MA, August 2000.

[Cover&Thomas 1991] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* John Wiley & Sons, Inc., New York, NY, 1991.

[Doucet et al. 2000] A. Doucet, J.F.G. de Freitas, and N.J. Gordon. *Sequential Monte-Carlo Methods in Practice*. Springer-Verlag, 2000.

[Estrin et al. 1999] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, Next Century Challenges: scalable coordination in sensor networks. In Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99), Seattle, Washington, August 1999.

[Gao et al. 2001] J. Gao, L.J. Guibas, J. Hershburger, L. Zhang, and A. Zhu. Discrete Mobile Centers. Proc. of 17th Symposium on Computational Geometry, Tufts University, MA, June, 2001.

[Geman&Jedynak 1996] D. Geman and B. Jedynak, An active testing model for tracking roads from satellite images, *IEEE Trans. Pattern Anal. Mach. Intell.*, 18, 1-14, 1996.

[Huang&Garcia-Molina 2001] Y. Huang and H. Garcia-Molina, Publish/Subscribe in a Mobile Environment. Proc. MobiDE-01, 2001.

[Intanagonwiwat et al. 2000] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM 2000), Boston, Massachusetts, August 2000.

[Manyika&Durrant-Whyte] J. Manyika and H. Durrant-Whyte, *Data fusion and sensor management: a decentralized information-theoretic approach.* Ellis Horwood, 1994.

[Mutambara 1998] A.G.O. Mutambara, *Decentralized estimation and control for multi-sensor systems.* CRC Press, 1998.

[Nakamura 1992] Y. Nakamura, Geometric Fusion: Minimizing uncertainty ellipsoid volumes. In *Data Fusion in Robotics and Machine Intelligence*, M.A. Abidi and R.C. Conzalez (eds), Academic Press, 1992.

[Pottie&Kaiser 2000] G.J. Pottie and W.J. Kaiser, "Wireless integrated network sensors." *Comm. ACM*, 43(5):51-58, May 2000.

[Singh et al 1998] S. Singh, M. Woo, and C.S. Raghavendra, Power-aware routing in mobile ad hoc networks. Proc. MOBICOM'98, 1998.

[Sohrabi 2000] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, Protocols for self-organization of a wireless sensor network. IEEE Personal Communications, 7(5):16-27, Oct 2000.

A Measurement Likelihood Function Derivation

Based on the measurement model (11) of Section 4, a closed form expression for the likelihood can be derived as

$$
\begin{aligned}
& p(z_1, \ldots z_N \mid \mathbf{x}) \\
= \quad & p\left(\left\{ w_i = z_i - \frac{a}{d_i^{\frac{\alpha}{2}}} \right\}_{i=1}^{N} \mid \mathbf{x}\right) \\
= \quad & \int_{a_{low}}^{a_{high}} p\left(a = y, \left\{ w_i = z_i - \frac{y}{d_i^{\frac{\alpha}{2}}} \right\}_{i=1}^{N} \mid \mathbf{x}\right) dy \\
= \quad & \int_{a_{low}}^{a_{high}} p(a = y) \prod_{i=1}^{N} p\left(w_i = z_i - \frac{y}{d_i^{\frac{\alpha}{2}}}\right) dy \\
= \quad & \frac{\sqrt{D(\mathbf{x})} e^{-\frac{1}{2}(C - B^2(\mathbf{x}) D(\mathbf{x}))}}{\Delta a (2\pi)^{\frac{N}{2} - \frac{1}{2}} \prod_{i=1}^{N} \sigma_i} \times \\
& \left[\Phi\left(\frac{a_{high} - B(\mathbf{x}) D(\mathbf{x})}{\sqrt{D(\mathbf{x})}} \right) - \right. \\
& \left. \Phi\left(\frac{a_{low} - B(\mathbf{x}) D(\mathbf{x})}{\sqrt{D(\mathbf{x})}} \right) \right]
\end{aligned}
$$

where

$$
B(\mathbf{x}) = \sum_{i=1}^{N} \frac{z_i}{\sigma_i^2 d_i^{\frac{\alpha}{2}}(\mathbf{x})} ,
$$

$$
C = \sum_{i=1}^{N} \frac{z_i^2}{\sigma_i^2} ,
$$

$$
\frac{1}{D(\mathbf{x})} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2 d_i^{\alpha}(\mathbf{x})} ,
$$

$$
\Delta a = a_{high} - a_{low} ,
$$

and

$$
d_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i\|
$$

$$
\Phi(x) = \frac{1}{2\pi} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt .
$$

Assuming that σ_i, α, a_{low}, a_{high}, and $\mathbf{x}_i$ are given *a priori*, the likelihood function (viewed as a function of $\mathbf{x}$) is parameterized by the measurement values $z_1, \ldots, z_N$.